Efficient implementation of Gauss collocation and Hamiltonian Boundary Value Methods

Luigi Brugnano[*] Gianluca Frasca Caccia[†] Felice Iavernaro[‡]

Devoted to celebrate the 80^{th} birthday of John Butcher

Abstract

In this paper we define an efficient implementation for the family of low-rank energy-conserving Runge-Kutta methods named Hamiltonian Boundary Value Methods (HBVMs), recently defined in the last years. The proposed implementation relies on the particular structure of the Butcher matrix defining such methods, for which we can derive an efficient splitting procedure. The very same procedure turns out to be automatically suited for the efficient implementation of Gauss-Legendre collocation methods, since these methods are a special instance of HBVMs. The linear convergence analysis of the splitting procedure exhibits excellent properties, which are confirmed by a few numerical tests.

Keywords: Energy-conserving methods; Hamiltonian Boundary Value Methods; W-transform; Gauss-Legendre collocation methods; Implicit Runge-Kutta methods; Splitting.

MSC (2010): 65P10, 65L05, 65L06, 65L99.

1 Introduction

The efficient numerical solution of implicit Runge-Kutta methods has been the subject of many investigations in the last decades, starting from the seminal papers of Butcher [23, 24] (see also [25]). This aspect is even more relevant when dealing with *geometric* Runge-Kutta methods, that is, methods used in the framework of *Geometric Integration* where, usually, the discrete problems generated by the methods need to be solved to within full machine accuracy, in order not to waste the specific properties of the methods.

In more details, in this paper we shall deal with the numerical solution of Hamiltonian problems, namely problems in the form,

$$y' = J\nabla H(y), \qquad y(t_0) = y_0 \in \mathbb{R}^{2m}, \tag{1}$$

[*] Dipartimento di Matematica e Informatica "U. Dini", Università di Firenze, Italy (luigi.brugnano@unifi.it)

[†] Dipartimento di Matematica e Informatica "U. Dini", Università di Firenze, Italy (frasca@math.unifi.it)

[‡] Dipartimento di Matematica, Università di Bari, Italy (felix@dm.uniba.it)

where

$$y = \begin{pmatrix} q \\ p \end{pmatrix}, \quad q, p \in \mathbb{R}^m, \qquad J = \begin{pmatrix} O & I_m \\ -I_m & O \end{pmatrix}, \tag{2}$$

$H(y)$ is the (scalar) *Hamiltonian* function defining the problem, and I_m the identity matrix of dimension m.[1] Due to the skew-symmetry of matrix J one has

$$\frac{\mathrm{d}}{\mathrm{d}t}H(y(t)) = \nabla H(y(t))^T y'(t) = \nabla H(y(t))^T J \nabla H(y(t)) = 0,$$

so that

$$H(y(t)) = H(y_0), \qquad t \ge t_0.$$

For isolated mechanical systems, the Hamiltonian has the physical meaning of the total energy of the system, so that often the Hamiltonian is referred to as the *energy*. Its conservation is, therefore, a significant feature for the discrete dynamical system induced by a numerical method for solving (1): methods having this property are usually called *energy-conserving methods*. Among such methods, we are interested in the class of energy-conserving methods named *Hamiltonian Boundary Value Methods (HBVMs)* [9, 10, 11, 8, 12, 13, 14, 15, 16, 4, 5, 6, 3], which have been recently devised starting from the idea of *discrete line integrals*, defined in [29, 30, 32] (see also [31]). For such methods, the discrete problem can be conveniently posed in a suitable form which can be exploited to derive efficient implementation strategies, as was done in [12]. Here we further improve on such results, by proposing and analysing an iterative procedure based on the particular structure of the discrete problem. As a by product, an efficient implementation of Gauss-Legendre collocation methods is also obtained. Indeed, these latter methods may be interpreted as a particular instance of HBVMs. The proposed procedure is strictly related to that recently devised in [7] for Radau IIA collocation formulae, though the two approaches are substantially different.

With this premise, the paper is organized as follows: in Section 2 we describe the structure of the discrete problem generated by HBVMs, along with the way of solving it, as done so far; in Section 3 we introduce the new iterative procedure, which is based on a suitable splitting; in Section 4 we study the convergence properties of the new iteration, also comparing it with known existing ones; in Section 5 a few numerical tests are reported; at last, a few conclusions are contained in Section 6.

2 Discrete problem induced by HBVMs

We now recall the basic facts about HBVMs, and derive the most efficient formulation of the generated discrete problems. Let us assume, for sake of brevity, $t_0 = 0$ in (1), and consider the approximation of the problem over the interval $[0, h]$, which will clearly concern the very first application of a given numerical method. Let us then consider the orthonormal polynomial basis, on the interval $[0, 1]$, provided by the shifted and scaled Legendre polynomials $\{P_j\}$:

$$\deg P_i = i, \qquad \int_0^1 P_i(x)P_j(x)\mathrm{d}x = \delta_{ij}, \qquad \forall i, j \ge 0, \tag{3}$$

[1] In the following, when the size of the identity is not specified, it can be deduced from the context.

where δ_{ij} is the usual Kronecker symbol. Under suitable mild assumptions on the Hamiltonian function H, the right-hand side of the differential equation (1) can be expanded along the considered basis, thus giving

$$y'(ch) = \sum_{j\ge0} \gamma_j(y) P_j(c), \qquad c \in [0,1], \tag{4}$$

where

$$\gamma_j(y) = \int_0^1 J\nabla H(y(\tau h)) P_j(\tau)\mathrm{d}\tau, \qquad j \ge 0. \tag{5}$$

By imposing the initial condition, the solution of this problem is formally obtained by

$$y(ch) = y_0 + h\sum_{j\ge0} \gamma_j(y) \int_0^c P_j(x)\mathrm{d}x, \qquad c \in [0,1]. \tag{6}$$

In order to derive a polynomial approximation σ of degree s to (6), we consider the following approximated ODE-IVPs:

$$\sigma'(ch) = \sum_{j=0}^{s-1} \gamma_j(\sigma) P_j(c), \qquad c \in [0,1], \qquad \sigma(0) = y_0, \tag{7}$$

where $\gamma_j(\sigma)$ is defined according to (5), by formally replacing y by σ. Consequently, the approximation to (6) will be given by

$$\sigma(ch) = y_0 + h\sum_{j=0}^{s-1} \gamma_j(\sigma) \int_0^c P_j(x)\mathrm{d}x, \qquad c \in [0,1]. \tag{8}$$

Assume now that the Hamiltonian function is a polynomial of degree ν. Consequently, the (unknown) vector coefficients $\{\gamma_j(\sigma)\}$ can be exactly obtained by using a quadrature formula defined at the Gaussian abscissae $\{c_1, \dots, c_k\}$, i.e.,

$$P_k(c_i) = 0, \qquad i = 1, \dots, k, \tag{9}$$

and corresponding weights $\{b_1, \dots, b_k\}$,[2]

$$\gamma_j(\sigma) = \sum_{i=1}^{k} b_i J\nabla H(\sigma(c_i h)) P_j(c_i), \qquad j = 0, \dots, s-1, \tag{10}$$

provided that

$$\nu \le \frac{2k}{s}. \tag{11}$$

By setting

$$Y_i = \sigma(c_i h), \qquad i = 1, \dots, k, \tag{12}$$

[2] Hereafter, we shall always assume this choice.

and considering that the new approximation is given by

$$y_1 \equiv \sigma(h) = y_0 + h\int_0^1 J\nabla H(\sigma(\tau h))\mathrm{d}\tau \equiv y_0 + h\sum_{i=1}^k b_i J\nabla H(Y_i),$$

one then obtains the following k-stage Runge-Kutta method,

$$\begin{array}{c|c} \boldsymbol{c} & A \\ \hline & \boldsymbol{b}^T \end{array} \tag{13}$$

where, as usual, $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^k$ are the vectors containing the weights and the abscissae, respectively, and (see, e.g. [12, 13, 14])

$$A = \mathcal{P}_{s+1}\hat{X}_s\mathcal{P}_s^T\Omega \in \mathbb{R}^{k\times k}, \tag{14}$$

with

$$\mathcal{P}_r = \begin{pmatrix} P_0(c_1) & \dots & P_{r-1}(c_1) \\ \vdots & & \vdots \\ P_0(c_k) & \dots & P_{r-1}(c_k) \end{pmatrix} \in \mathbb{R}^{k\times r}, \qquad r = s, s+1, \tag{15}$$

$$\hat{X}_s = \left(\begin{array}{cccc} \frac{1}{2} & -\xi_1 & & \\ \xi_1 & 0 & \ddots & \\ & \ddots & \ddots & -\xi_{s-1} \\ & & \xi_{s-1} & 0 \\ \hline & & & \xi_s \end{array}\right) \equiv \left(\begin{array}{c} X_s \\ \hline 0\dots 0\,\xi_s \end{array}\right) \in \mathbb{R}^{s+1\times s}, \tag{16}$$

$$\xi_i = \left(2\sqrt{4i^2-1}\right)^{-1}, \qquad i = 1, \dots, s, \tag{17}$$

$$\Omega = \mathrm{diag}(\boldsymbol{b}) \in \mathbb{R}^{k\times k}. \tag{18}$$

According to [11], we give the following definition (see also [9]).

Definition 1 *(13)–(18) define a HBVM(k,s) method.*

The following properties [11, 14] elucidate the role of two indices k (number of ascissae) and s (degree of the underlying polynomial σ) in the previous definition.

Theorem 1 *For all $k \ge s$, a HBVM(k,s) method:*

- *has order $2s$, that is:*
$$y_1 - y(h) = O(h^{2s+1});$$

- *is energy conserving for all polynomial Hamiltonians of degree ν satisfying (11);*

- *for general non-polynomial (but suitably regular) Hamiltonians, one has:*

$$H(y_1) - H(y_0) = O(h^{2k+1}). \tag{19}$$

Remark 1 *From (19) one deduces that a HBVM(k,s) method is* practically *energy-conserving also in the case of non-polynomial Hamiltonians, provided that k is large enough. Indeed, it is enough to approximate the involved integrals to within round-off errors.*

For sake of completeness, and for later reference, we also prove the following result, which actually shows that HBVM(k,s) methods can be regarded as a generalization of the s-stage Gauss-Legendre collocation formulae [11].

Theorem 2 *HBVM(s,s) coincides with the s-stage Gauss-Legendre collocation method.*

Proof. When $k=s$, from (9) and (14)–(18) one obtains:

$$\mathcal{P}_{s+1} = \left(\begin{array}{cc} \mathcal{P}_s & \mathbf{0} \end{array}\right) \qquad \Rightarrow \qquad A = \mathcal{P}_s X_s \mathcal{P}_s^T \Omega.$$

Moreover,

$$\mathcal{P}_s^T \Omega \mathcal{P}_s = I_s,$$

since, by setting $\boldsymbol{e}_i$ the ith unit vector,

$$\boldsymbol{e}_i^T \left(\mathcal{P}_s^T \Omega \mathcal{P}_s\right) \boldsymbol{e}_j = \sum_{\ell=1}^k b_\ell P_{i-1}(c_\ell) P_{j-1}(c_\ell) = \int_0^1 P_{i-1}(x) P_{j-1}(x) \mathrm{d}x = \delta_{ij},$$

due to the fact that the quadrature is exact, and because of (3). Consequently,

$$\mathcal{P}_s^T \Omega = \mathcal{P}_s^{-1}$$

and, then,

$$A = \mathcal{P}_s X_s \mathcal{P}_s^{-1}, \tag{20}$$

which is the W-transformation [26] of the s-stage Gauss-Legendre collocation method. □

If we set $\boldsymbol{y}$ the (block) vector with the internal stages (12) and $\boldsymbol{e} = (1,\ldots,1)^T \in \mathbb{R}^k$, the discrete problem generated by a HBVM(k,s) method is given by

$$\boldsymbol{y} = \boldsymbol{e} \otimes y_0 + hA \otimes J\,\nabla H(\boldsymbol{y}), \tag{21}$$

which is a nonlinear system of (block) dimension k.[3] However, in view of (19), k is likely to be much higher than s and, consequently, such a formulation is in general not recommendable.

To derive a more efficient formulation, let us set $\boldsymbol{\gamma}$ the (block) vector containing the coefficients defining the polynomial σ in (8), thus obtaining:

$$\boldsymbol{\gamma} = \mathcal{P}_s^T \Omega \otimes J\,\nabla H(\boldsymbol{y}), \qquad \boldsymbol{y} = \boldsymbol{e} \otimes y_0 + h\mathcal{P}_{s+1}\hat{X}_s \otimes I\,\boldsymbol{\gamma}.$$

Combined together, such equations provide us with the following discrete problem,

$$F(\boldsymbol{\gamma}) \,\equiv\, \boldsymbol{\gamma} - \mathcal{P}_s^T \Omega \otimes J\,\nabla H\left(\boldsymbol{e} \otimes y_0 + h\mathcal{P}_{s+1}\hat{X}_s \otimes I\,\boldsymbol{\gamma}\right) = \mathbf{0}, \tag{22}$$

[3] Here $\nabla H(\boldsymbol{y})$ is the block vector whose entries are given by the gradient of H evaluated at the k stages.

whose (block) size is always s, independently of k. One easily checks that the simplified Newton iteration, applied for solving (22) consists in the following iteration [12]:

$$\begin{aligned} \text{solve}: \ \left[I - hX_s \otimes J\nabla^2 H_0\right] \Delta^\ell &= -F(\boldsymbol{\gamma}^\ell) \\ \boldsymbol{\gamma}^{\ell+1} &= \boldsymbol{\gamma}^\ell + \Delta^\ell, \qquad \ell = 0, 1, \dots, \end{aligned} \tag{23}$$

where $\nabla^2 H_0$ is the Hessian of $H(y)$ evaluated at y_0. Consequently, the bulk of the computational cost is due to the factorization of the matrix

$$M_0 = I - hX_s \otimes J\nabla^2 H_0,$$

having dimension $2sm \times 2sm$. In the next section, we shall see how to efficiently solve the iteration (23).

3 The new splitting procedure

The iteration (23) is similar in structure to the simplified-Newton iteration applied to the original system (21), for which a number of splitting procedures have been devised (see, e.g., [27, 28, 1, 2, 7, 17, 18, 21, 19, 20]). However, the triangular splitting iteration defined in [27, 28], and the modified triangular splitting iteration defined in [1] turn out to be not effective for (23), due to the particular structure of the matrix X_s (see (16)). Conversely, the *blended iteration* defined in [2, 17, 18, 21, 19, 20] turns out more appropriate, as is shown in [12]. We here shall devise a different iterative procedure, which appears to be even more favourable. This is the subject of the remaining part of this section. The main idea is similar to that devised in [7] for Radau IIA collocation methods, even though the framework and the overall details (and results) are different.

To begin with, let us consider the polynomial

$$\hat{\gamma}(c) = \sum_{j=0}^{s-1} P_j(c)\gamma_j(\sigma), \qquad c \in \mathbb{R}, \tag{24}$$

and introduce a new set of (block) unknowns,

$$\hat{\gamma}_i \equiv \sum_{j=0}^{s-1} P_j(\hat{c}_i)\gamma_j(\sigma), \qquad i = 1, \dots, s, \tag{25}$$

defined as the evaluation of (24) at the set of *auxiliary abscissae*

$$\hat{c}_1 < \cdots < \hat{c}_s. \tag{26}$$

Introducing the (block) vector

$$\hat{\boldsymbol{\gamma}} = \begin{pmatrix} \hat{\gamma}_1 \\ \vdots \\ \hat{\gamma}_s \end{pmatrix}, \tag{27}$$

and the matrix

$$\hat{\mathcal{P}} = \left(\ P_{j-1}(\hat{c}_i)\ \right) \in \mathbb{R}^{s\times s}, \tag{28}$$

we can recast (25) in matrix notation as

$$\hat{\boldsymbol{\gamma}} = \hat{\mathcal{P}} \otimes I\, \boldsymbol{\gamma}. \tag{29}$$

In terms of the new unknown vector $\hat{\boldsymbol{\gamma}}$, the simplified Newton iteration (23) reads:

$$\begin{array}{rcl} \text{solve} : \hat{M}_0 \hat{\Delta}^\ell &=& -\hat{\mathcal{P}} \otimes I\, F(\hat{\mathcal{P}}^{-1} \otimes I\, \hat{\boldsymbol{\gamma}}^\ell) \equiv \boldsymbol{\eta}^\ell \\ \hat{\boldsymbol{\gamma}}^{\ell+1} &=& \hat{\boldsymbol{\gamma}}^\ell + \hat{\Delta}^\ell, \qquad \ell = 0, 1, \ldots, \end{array} \tag{30}$$

where

$$\hat{M}_0 = I - h\left(\hat{\mathcal{P}} X_s \hat{\mathcal{P}}^{-1}\right) \otimes J\nabla^2 H_0 \equiv I - h\hat{A} \otimes J\nabla^2 H_0. \tag{31}$$

The key idea is that of choosing the abscissae (26) such that the Crout factorization of $\hat{A}$

$$\hat{A} = \hat{L}\hat{U}, \tag{32}$$

has the diagonal entries of $\hat{L}$ all equal. In such a case, by following the approach of van der Houwen et al. [27, 28], the iteration (30) is replaced by the *inner-outer* iteration

$$\begin{array}{rcl} \text{solve} : \left[I - h\hat{L} \otimes J\nabla^2 H_0\right] \hat{\Delta}^{\ell,r+1} &=& h\hat{L}(\hat{U} - I) \otimes J\nabla^2 H_0 \hat{\Delta}^{\ell,r} + \boldsymbol{\eta}^\ell, \\ && r = 0, 1, \ldots, \nu - 1, \\ \hat{\boldsymbol{\gamma}}^{\ell+1} &=& \hat{\boldsymbol{\gamma}}^\ell + \hat{\Delta}^{\ell,\nu}, \qquad \ell = 0, 1, \ldots. \end{array} \tag{33}$$

In particular, since $\hat{\Delta}^{\ell,0} = 0$, the choice $\nu = 1$ corresponds to the approach used by van der Houwen et al. to devise PTIRK methods [27], whereas, if ν is large enough to have full convergence of the inner-iteration (the one on r), then the outer iteration is equivalent to (30). Clearly, all the intermediate possibilities can be suitably considered.

That the diagonal entries of the factor $\hat{L}$ are all equal to a given value, say d_s, has the obvious advantage that one only needs to factor the matrix

$$I - hd_s J\nabla^2 H_0 \in \mathbb{R}^{2m\times 2m}.$$

Concerning d_s, the following result holds true.

Theorem 3 *Assume that the factorization (32) is defined and that the factor $\hat{L}$ has all its diagonal entries equal to d_s. Then, with reference to (17), one has:*

$$d_s = \begin{cases} \sqrt[s]{\prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i-1}^2}\,, & \textit{if } s \textit{ is even}, \\ \sqrt[s]{\frac{1}{2}\prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i}^2}\,, & \textit{if } s \textit{ is odd}. \end{cases} \tag{34}$$

Proof. Assume that (31)–(32) hold true. Then

$$\det(X_s) = \det(\hat{\mathcal{P}} X_s \hat{\mathcal{P}}^{-1}) = \det(\hat{A}) = \det(\hat{L}\hat{U}) = \det(\hat{L}) = d_s^s,$$

since $\hat{U}$ has unit diagonal and all the entries of $\hat{L}$ are equal to d_s. Consequently,

$$d_s = \sqrt[s]{\det(X_s)}.$$

The thesis then follows by considering that, from (16),

$$\det(X_1) = \frac{1}{2}, \qquad \det(X_2) = \xi_1^2,$$

and, by and applying the Laplace formula, one obtains:

$$\det(X_s) = \begin{cases} \prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i-1}^2\,, & \text{if } s \text{ is even,} \\ \frac{1}{2} \prod_{i=1}^{\lfloor \frac{s}{2} \rfloor} \xi_{2i}^2\,, & \text{if } s \text{ is odd.} \end{cases} \qquad (35)$$

□

By virtue of the previous result, in order to compute the auxiliary abscissae (26), we have symbolically solved the following set of equations, which is obviously equivalent to requiring that the factor $\hat{L}$ has the diagonal entries equal to each other:

$$\det(\hat{A}_{\ell+1}) = d_s \det(\hat{A}_\ell), \qquad \ell = 1, \ldots, s-1. \qquad (36)$$

Here $\hat{A}_\ell$ denotes the principal leading submatrix of order ℓ of $\hat{A}$, and d_s is given by (34).

Remark 2 *We observe that the auxiliary abscissae (26) are s, whereas the conditions (36) are $s-1$. This means that a further condition can be imposed on the abscissae. We have chosen it in order to (approximately) minimize the* maximum amplification factor *of the iteration, as described in Section 4.*

The obtained results are listed in Table 1, for $s = 2, \ldots, 6$, from which one sees that in all cases the abscissae are distinct and inside the interval $[0, 1]$.

We emphasize that, for any given s, the distribution of the abscissae $\hat{c}_i$ is independent of k and so is the factorization (32) of the matrix $\hat{A}$ whose computation is responsible of the bulk of the computational effort during the integration process. This property has a relevant consequence during the implementation phase of this class of methods. In fact, one can conjecture a procedure to advance the time that dynamically selects the most appropriate value of k. Depending on the specific problem at hand and the configuration of the system at the given time, one can easily switch from a symplectic to an energy preserving method by choosing $k = s$ (Gauss method) or $k > s$, respectively.

4 Convergence analysis and comparisons

In this section we briefly analyze the splitting procedure (33) according to the linear analysis of convergence in [27] (see also [20]). In such a case, problem (1) becomes the celebrated test equation

$$y' = \lambda y, \qquad y(t_0) = y_0. \qquad (37)$$

Table 1: Auxiliary abscissae (26) for the HBVM(k, s) and s-stage Gauss method, $s = 2, \dots, 6$, and the diagonal entry d_s (see (34)) of the corresponding factor $\hat{L}$.

$s = 2$	
$\hat{c}_1$	0.26036297108184508789101036587842555
$\hat{c}_2$	1
d_2	0.28867513459481288225457439025097873
$s = 3$	
$\hat{c}_1$	0.15636399930006671060146617869938122
$\hat{c}_2$	0.45431868644630821020177903150137523
$\hat{c}_3$	0.948
d_3	0.20274006651911333949661483325792675
$s = 4$	
$\hat{c}_1$	0.10980739789315927030006609946548674
$\hat{c}_2$	0.33170147176242312021431664090897753
$\hat{c}_3$	0.34895222925320098805117573706272031
$\hat{c}_4$	0.7186
d_4	0.15619699684601279005430416526875577
$s = 5$	
$\hat{c}_1$	0.084665152651785050300050089866691763
$\hat{c}_2$	0.257176246353319497655094082620443041
$\hat{c}_3$	0.406997899472141237569051120583722535
$\hat{c}_4$	0.543482236497480925424020865878224100
$\hat{c}_5$	0.8858
d_5	0.12702337351164258963093490787943281
$s = 6$	
$\hat{c}_1$	0.067421939209393759076155059271608432
$\hat{c}_2$	0.199433361269530107247831326370752698
$\hat{c}_3$	0.375498823199182884162061752054007488
$\hat{c}_4$	0.44562087499073615218681054513866021
$\hat{c}_5$	0.65709214779914978056665114753638658
$\hat{c}_6$	1
d_6	0.10702845478806509529222890981996019

By setting, as usual, $q = h\lambda$, one then obtains that the error equation associated with the outer iteration (33) is given by

$$e_{\ell+1} = Z(q)^\nu e_\ell, \qquad Z(q) := q(I - q\hat{L})^{-1}\hat{L}(\hat{U} - I), \qquad \ell = 0, 1, \dots, \tag{38}$$

where $e_\ell = \hat{\Delta}^{\ell,\nu} - \hat{\Delta}$ is the error vector at step ℓ and $Z(q)$ is the iteration matrix induced by the splitting procedure. This latter will converge if and only if its spectral radius,

$$\rho(q) := \rho(Z(q)),$$

is less than 1. The *region of convergence* of the iteration is then defined as

$$\mathbb{D} = \{q \in \mathbb{C} \,:\, \rho(q) < 1\}\,.$$

The iteration is said to be *A-convergent* if $\mathbb{C}^- \subseteq \mathbb{D}$. If, in addition, the *stiff amplification factor*,

$$\rho^\infty := \lim_{q \to \infty} \rho(q),$$

is null, then the iteration is said to be *L-convergent.* Clearly, A-convergent iterations are appropriate for A-stable methods, and L-convergent iterations are appropriate for L-stable methods. In our case, since

$$Z(q) \to (\hat{U} - I), \qquad q \to \infty, \tag{39}$$

which is a nilpotent matrix of index s, the iteration is L-convergent if and only if it is A-convergent. Since the iteration is well defined for all $q \in \mathbb{C}^-$ (due to the fact that the diagonal entry of $\hat{L}$, d_s, is positive, as was shown in (34)) and $\rho(0) = 0$, A-convergence, in turn, is equivalent to require that the *maximum amplification factor*,

$$\rho^* = \max_{x \in \mathbb{R}} \rho(ix),$$

is not larger than 1. Another useful parameter is the *nonstiff amplification factor*,

$$\tilde{\rho} = \rho(\hat{L}(\hat{U} - I)), \tag{40}$$

that governs the convergence for small values of q since

$$\rho(q) \approx \tilde{\rho}|q|, \qquad \text{for} \quad q \approx 0.$$

Clearly, the smaller ρ^* and $\tilde{\rho}$, the better the convergence properties of the iteration.

Remark 3 *As anticipated in Remark 2, the additional condition imposed on the auxiliary abscissae (26) is that of (approximately) minimizing the maximum amplification factor ρ^* of the iteration, while fulfilling the conditions (36).*

In Table 2 we list the nonstiff amplification factors and the maximum amplification factors for the following L-convergent iterations applied to the s-stage Gauss-Legendre methods:

(i) the iteration obtained by the original triangular splitting in [27];

Table 2: Amplification factors for the triangular splitting in [27], the modified triangular splitting in [1], the *blended* iteration in [17], and the splitting (33), for the s-stage Gauss-Legendre formulae. The last two cases coincide with those for the HBVM(k, s) methods, $k \geq s$.

	(i): triangular splitting in [27]		(ii): triangular splitting in [1]		(iii): *blended* iteration in [17]		(iv): triangular splitting (30)	
s	$\tilde{\rho}$	ρ^*	$\tilde{\rho}$	ρ^*	$\tilde{\rho}$	ρ^*	$\tilde{\rho}$	ρ^*
2	0.1429	0.0833	0.1340	0.0774	0.1340	0.0774	0.1340	0.0774
3	0.3032	0.1098	0.2537	0.0856	0.2765	0.1088	0.2536	0.0870
4	0.4351	0.1126	0.3492	0.0803	0.3793	0.1119	0.2656	0.0696
5	0.5457	0.1058	0.4223	0.0730	0.4544	0.1066	0.3275	0.0534
6	0.6432	0.0973	0.4861	0.0702	0.5114	0.0993	0.4243	0.0878

(ii) the iteration obtained by the modified triangular splitting in [1];

(iii) the *blended* iteration obtained by the *blended implementation* of the methods, as defined in [17];

(iv) the iteration defined by (33).

We recall that the scheme (i) (first column) requires s real factorizations per iteration, whereas (ii)–(iv) only need one factorization per iteration. From the parameters listed in the table, one concludes that the proposed splitting procedure is the most effective among all the considered ones.

Remark 4 *For sake of accuracy, we stress that, when dealing with the actual implementation of HBVM(k, s) methods, only the blended iteration [12] and the newly proposed one (33) can be considered, whereas the triangular splitting defined in [27] and its modified version [1] turn out to be not effective, as was pointed out at the beginning of Section 3. Consequently, in such a case, one has to consider only the last two group of columns in Table 2.*

5 Numerical Tests

In this section, we report a numerical example taken from [12], namely the Hamiltonian problem defined by the Hamiltonian

$$H(q, p) = p^2 + \beta^2 q^2 + \alpha^2 (q + p)^{2n} \tag{41}$$

with the following parameters (slightly modified, with respect to those considered in [12]):

$$\beta = 12, \qquad \alpha = 4, \qquad n = 4. \tag{42}$$

Table 3: Number of fixed-point, blended iterations, and splitting iterations (– means no convergence) required for each level curve of problem (41)–(43), by using the HBVM(8,2) and HBVM(12,3) methods with stepsize $h = 10^{-3}$, along with the maximum error in the numerical Hamiltonian.

	HBVM(8,2)				HBVM(12,3)			
i	Fixed-point iterations	Blended iterations	Splitting iteration	errH	Fixed-point iterations	Blended iterations	Splitting iteration	errH
0	9191	6957	4814	$2.9 \cdot 10^{-15}$	8544	7398	4681	$3.1 \cdot 10^{-15}$
1	16835	12151	8648	$1.6 \cdot 10^{-14}$	14684	12486	8086	$7.4 \cdot 10^{-15}$
2	23907	16344	12293	$6.6 \cdot 10^{-14}$	19004	16860	11095	$3.2 \cdot 10^{-14}$
3	33396	21854	17089	$3.2 \cdot 10^{-14}$	23854	22554	15415	$9.8 \cdot 10^{-14}$
4	50163	30762	23744	$5.4 \cdot 10^{-14}$	31263	31463	22445	$9.3 \cdot 10^{-14}$
5	–	48109	35994	$6.5 \cdot 10^{-14}$	42598	45747	34885	$1.5 \cdot 10^{-13}$
6	–	–	51961	$9.9 \cdot 10^{-13}$	–	71105	49239	$2.9 \cdot 10^{-13}$
7	–	–	–	–	–	–	66500	$2.3 \cdot 10^{-13}$

Consequently, the Hamiltonian is a polynomial of degree $\nu = 8$ and, from (11), one has that HBVM$(8, 2)$ and HBVM$(12, 3)$ methods are energy-conserving, with orders 4 and 6, respectively. We integrate the orbits starting at the initial points

$$\left(\; q_0, \quad p_0 \; \right) = \left(\; i + 0.1, \quad -(i + 0.1) \; \right), \qquad i = 0, ..., 7. \tag{43}$$

These orbits are closed (see Figure 1) and in [12] it has been shown that, for their numerical integration, energy-conserving methods are more appropriate than symplectic integrators (like Gauss-Legendre collocation methods). We use a constant stepsize $h = 10^{-3}$ for integrating the eight trajectories for 10^3 steps. In Table 3 we list the results obtained by the two methods, in terms of required iterations for solving the generated discrete problems, when using:

- a fixed-point iteration;
- a blended iteration;
- the splitting iteration (33) with 2 inner iterations.

The choice of 2 inner iterations in (33) makes the cost of one outer iteration comparable to that of one blended iteration, provided that (33) is implemented as suggested in [7].[4]

From Table 3 we see that the proposed splitting iteration (33) appears to be the most reliable among the considered ones and, in particular, it is competitive with the blended implementation of HBVMs. Similar results are obtained when considering Gauss-Legendre collocation methods, even though the qualitative behavior of the computed solution may be not correct when moving away from the origin (see [7]).

[4] The implementation details of the new procedure will be the object of a future work.

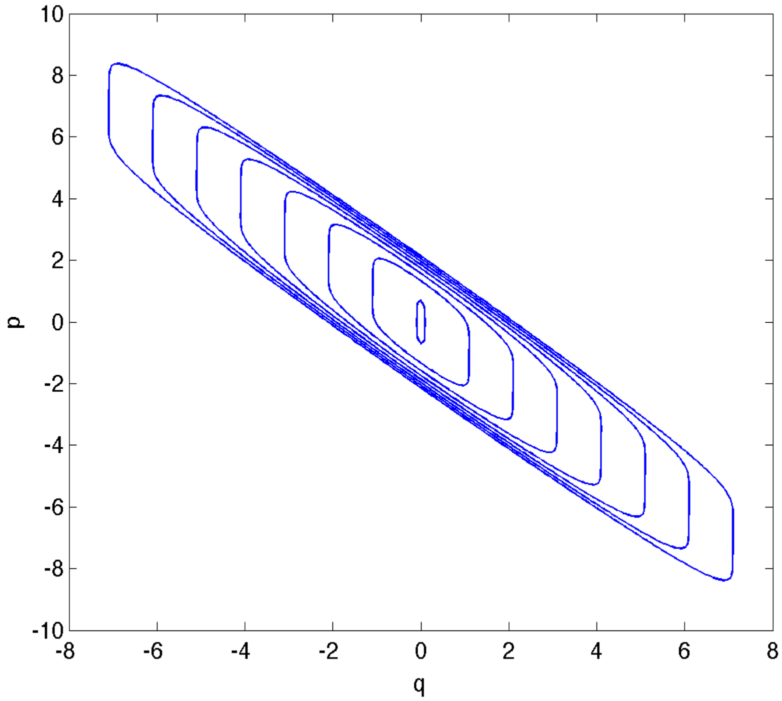

Figure 1: Level curves for problem (41)–(43).

6 Conclusions

In this paper we have defined an efficient iterative procedure for solving the discrete problems generated by the application of HBVM(k, s) methods, a class of energy-conserving methods for polynomial Hamiltonian dynamical systems. The proposed implementation turns out to improve over that proposed in [12], as is confirmed by a few numerical tests. In particular, it also applies to Gauss-Legendre formulae and the resulting linear convergence analysis shows that the proposed iterative procedure is the most effective, among those based on suitable splittings of the corresponding Butcher array of the methods.

References

[1] P. Amodio, L. Brugnano. A Note on the Efficient Implementation of Implicit Methods for ODEs. *J. Comput. Appl. Math.* **87** (1997) 1–9.

[2] L. Brugnano. Blended Block BVMs (B$_3$VMs): A Family of Economical Implicit Methods for ODEs. *J. Comput. Appl. Math.* **116** (2000) 41–62.

[3] L. Brugnano, M. Calvo, J.I. Montijano, L. Ràndez. Energy preserving methods for Poisson systems. *J. Comput. Appl. Math.* **236** (2012) 3890–3904.

[4] L. Brugnano, F. Iavernaro. Line Integral Methods which preserve all invariants of conservative problems. *J. Comput. Appl. Math.* **236** (2012) 3905–3919.

[5] L. Brugnano, F. Iavernaro. Geometric Integration by Playing with Matrices. *AIP Conf. Proc.* **1479** (2012) 16–19.

[6] L. Brugnano, F. Iavernaro. Recent Advances in the Numerical Solution of Conservative Problems. *AIP Conf. Proc.* **1493** (2012) 175–182.

[7] L. Brugnano, F. Iavernaro, C. Magherini. Efficient implementation of Radau collocation methods. (submitted for publication) 2012, `arXiv:1302.1037`

[8] L. Brugnano, F. Iavernaro, T. Susca. Numerical comparisons between Gauss-Legendre methods and Hamiltonian BVMs defined over Gauss points. *Monografias de la Real Academia de Ciencias de Zaragoza* **33** (2010) 95–112.

[9] L. Brugnano, F. Iavernaro, D. Trigiante. Analisys of Hamiltonian Boundary Value Methods (HBVMs) for the numerical solution of polynomial Hamiltonian dynamical systems, 2009. `arXiv:0909.5659v1`

[10] L. Brugnano, F. Iavernaro, D. Trigiante. Hamiltonian BVMs (HBVMs): a family of "drift-free" methods for integrating polynomial Hamiltonian systems. *AIP Conf. Proc.* **1168** (2009) 715–718.

[11] L. Brugnano, F. Iavernaro, D. Trigiante. Hamiltonian Boundary Value Methods (Energy Preserving Discrete Line Methods). *JNAIAM J. Numer. Anal. Ind. Appl. Math.* **5**,1-2 (2010) 17–37.

[12] L. Brugnano, F. Iavernaro, D. Trigiante. A note on the efficient implementation of Hamiltonian BVMs. *J. Comput. Appl. Math.* **236** (2011) 375–383.

[13] L. Brugnano, F. Iavernaro, D. Trigiante. The Lack of Continuity and the Role of Infinite and Infinitesimal in Numerical Methods for ODEs: the Case of Symplecticity. *Applied Mathematics and Computation* **218** (2012) 8053–8063.

[14] L. Brugnano, F. Iavernaro, D. Trigiante. A simple framework for the derivation and analysis of effective one-step methods for ODEs. *Applied Mathematics and Computation* **218** (2012) 8475–8485.

[15] L. Brugnano, F. Iavernaro, D. Trigiante. A two-step, fourth-order method with energy preserving properties. *Computer Physics Communications* **183** (2012) 1860–1868.

[16] L. Brugnano, F. Iavernaro, D. Trigiante. Energy and Quadratic Invariants–Preserving Integrators Based upon Gauss Collocation Formulae. *SIAM J. Numer. Anal.* **50**, 6 (2012) 2897-2916.

[17] L. Brugnano, C. Magherini. Blended Implementation of Block Implicit Methods for ODEs. *Appl. Numer. Math.* **42** (2002) 29–45.

[18] L. Brugnano, C. Magherini. The BiM Code for the Numerical Solution of ODEs. *J. Comput. Appl. Math.* **164-165** (2004) 145–158.

[19] L. Brugnano, C. Magherini. Blended Implicit Methods for solving ODE and DAE problems, and their extension for second order problems. *J. Comput. Appl. Math.* **205** (2007) 777–790.

[20] L. Brugnano, C. Magherini. Recent Advances in Linear Analysis of Convergence for Splittings for Solving ODE problems. *Appl. Numer. Math.* **59** (2009) 542–557.

[21] L. Brugnano, C. Magherini, F. Mugnai. Blended Implicit Methods for the Numerical Solution of DAE Problems. *J. Comput. Appl. Math.* **189** (2006) 34–50.

[22] K. Burrage, P.M. Burrage. Low-rank Runge-Kutta methods, symplecticity and stochastic Hamiltonian problems with additive noise. *J. Comput. Appl. Math.* **236** (2012) 3920–3930.

[23] J.C. Butcher. On the implementation of implicit Runge-Kutta methods. *BIT* **16** (1976) 237–240.

[24] J.C. Butcher. A transformed implicit Runge-Kutta method. *J. Assoc. Comput Mach.* **26** (1979) 237–240.

[25] G.J. Cooper, J.C. Butcher. An iteration scheme for implicit Runge-Kutta methods. *IMA J. Numer. Anal.* **3** (1983) 127–140.

[26] E. Hairer, G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Equations.* Springer, Berlin, 1991.

[27] P.J. van der Houwen, J.J.B. de Swart. Triangularly implicit iteration methods for ODE-IVP solvers. *SIAM J. Sci. Comput.* **18** (1997) 41–55.

[28] P.J. van der Houwen, J.J.B. de Swart. Parallel linear system solvers for Runge-Kutta methods. *Adv. Comput. Math.* **7**, 1-2 (1997) 157–181.

[29] F. Iavernaro, B. Pace. s-Stage Trapezoidal Methods for the Conservation of Hamiltonian Functions of Polynomial Type. *AIP Conf. Proc.* **936** (2007) 603–606.

[30] F. Iavernaro, B. Pace. Conservative Block-Boundary Value Methods for the Solution of Polynomial Hamiltonian Systems. *AIP Conf. Proc.* **1048** (2008) 888–891.

[31] F. Iavernaro, D. Trigiante. Discrete conservative vector fields induced by the trapezoidal method. *JNAIAM J. Numer. Anal. Ind. Appl. Math.* **1**, 1 (2006) 113–130.

[32] F. Iavernaro, D. Trigiante. High-order symmetric schemes for the energy conservation of polynomial Hamiltonian problems. *JNAIAM J. Numer. Anal. Ind. Appl. Math.* **4**, 1-2 (2009) 87–101.